

WHAT IS CLAIMED IS:

1. A Java run-time system comprising  
a stacked-based interpreter for executing a Java program comprising Java bytecode  
instructions and Java class structures;  
a converter for mapping standard Java symbolic linking strings contained in a  
downloaded Java program onto linking identifiers; and  
an export table for storing linking identifiers generated by the converter to bind a  
reference in a bytecode instruction to be executed to a corresponding link target.
2. A Java run-time system according to claim 1, wherein the converter is  
adapted to use a hash function to map the standard Java symbolic linking strings  
onto linking identifiers.
3. A Java run-time system according to claim 1, wherein the converter is  
adapted to use a parameterized hash function to map the standard Java symbolic  
linking strings onto linking identifiers, the parameter being contained in the Java  
program to be downloaded.
4. A Java development kit comprising a Java run-time system according to  
claim 3 and a Java conversion system for calculating a parameter for standard Java  
symbolic linking strings of a Java program to be downloaded on the Java run-time  
system.
5. A Java run-time system according to claim 1, said run-time system being  
ported on an embedded microcontroller of a smart card.
6. A Java run-time system according to claim 2, said run-time system being  
ported on an embedded microcontroller of a smart card.
7. A Java run-time system according to claim 3, said run-time system being  
ported on an embedded microcontroller of a smart card.
8. A method for downloading and linking a Java program on a Java run-time  
system comprising a stack-based interpreter for executing bytecode instructions,  
said Java program comprising Java bytecode instructions and Java class structures  
and including Java standard symbolic linking strings to bind a reference in a

bytecode instruction to be executed to a corresponding link target,  
comprising the steps of mapping the Java standard symbolic linking strings to  
linking identifiers; and storing said linking identifiers in an export table.

5           9.       A method according to claim 8, wherein a hash function is used to map the  
standard Java symbolic link strings onto linking identifiers.

10           10.       A method according to claim 8, wherein a parameterized hash function is  
used to map standard Java symbolic linking strings onto linking identifiers, said  
parameter being included in the Java program to be downloaded.

15           11.       A method according to claim 10, wherein the parameter for the Java  
program to be downloaded is used to ensure that the hash function does not map  
two symbolic linking strings of Java program to the same linking identifier.

20           12.       A program storage device readable by a computer, tangibly embodying a  
program of instructions executable by the computer to perform method steps of a  
method for downloading and linking a Java program on a Java run-time system  
comprising a stack-based interpreter for executing bytecode instructions, said Java  
program comprising Java bytecode instructions and Java class structures and  
including Java standard symbolic linking strings to bind a reference in a bytecode  
instruction to be executed to a corresponding link target, comprising the steps of  
mapping the Java standard symbolic linking strings to linking identifiers; and  
storing said linking identifiers in an export table.

25           13.       A program storage device according to claim 12, wherein a hash function is  
used to map the standard Java symbolic link strings onto linking identifiers.

30           14.       A program storage device according to claim 12, wherein a parameterized  
hash function is used to map standard Java symbolic linking strings onto linking  
identifiers, said parameter being included in the Java program to be downloaded.

35           15.       A program storage device according to claim 14, wherein the parameter for  
the Java program to be downloaded is used to ensure that the hash function does  
not map two symbolic linking strings of Java program to the same linking identifier.

16.       A program storage device according to claim 12, said computer comprising

an embedded microcontroller of a smart card.

1000959 : 44504  
10511 : 33630001